# logistic regression

1 Introduction

2 Interpretation of Hypothesis Output

3 Decision Boundary

4 Non-linear Decision Boundary

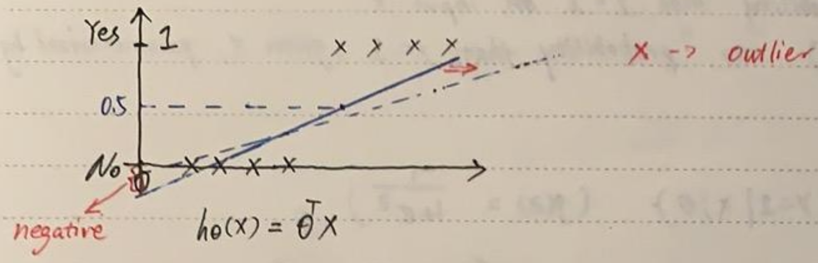
5 Cost Function

6 Gradient Descent

7 MLE

8 Regularization

# Logistic Regression 详细整理.




$h_\theta(x) = \theta^T X$

For linear regression $\Rightarrow$ Threshold classifier output $h_\theta(x)$ at 0.5

$$\text{Predict} = \begin{cases} 1 & \text{if } h_\theta(x) \geq 0.5 \\ 0 & \text{if } h_\theta(X) < 0.5 \end{cases}$$

Note: It's very sensitive to the outlier

It gives me negative prediction probability ( Don't make sense )

So, the problem is

for classification : $y = 0$ or $1$

$h_\theta(X)$ can be $> 1$ or $< 0$

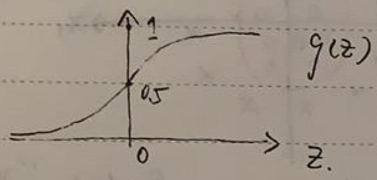
How can we fix it ???

We use logistic regression $\Longrightarrow$ we want $0 \leq h_\theta(X) \leq 1$

So, how to do that ???

$$\begin{cases} h_\theta(X) = g(\theta^T X) \\ g(z) = \dfrac{1}{1+e^{-z}} \end{cases} \longrightarrow \text{sigmoid function.}$$

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$



Interpretation of Hypothesis Output

$h_\theta(x)$ = estimated probability that $y=1$ on input $x$

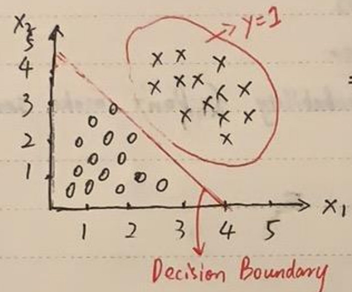
$= P(y=1 \mid x;\theta)$ => "probability that $y=1$, given $x$, parameterized by $\theta$"

Decision Boundary.

$h_\theta(x) = g(\theta^T x) = P(y=1 \mid x;\theta) \quad (g(z) = \frac{1}{1+e^{-z}})$

Suppose predict $\begin{cases} "y=1" \text{ if } h_\theta(x) \geq 0.5 \iff \theta^T x \geq 0 \\ "y=0" \text{ if } h_\theta(x) < 0.5 \iff \theta^T x < 0 \end{cases}$

Ex:



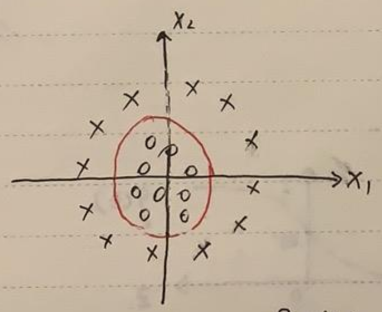

=> $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

After we fit the parameter, suppose we have

$\theta = \begin{bmatrix} -4 \\ 1 \\ 1 \end{bmatrix}$

Predict "$y=1$" if $\underbrace{-4 + x_1 + x_2}_{\theta^T x} \geq 0$

Non-linear decision boundaries.



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$

$\theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$

Predict "$y=1$" if $-1 + x_1^2 + x_2^2 \geq 0$

But if we get more complex or higher polynomial terms?

$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \cdots + \cdots)$

=> The shape of hyperplane will be more complex.

Cost function.

Ex. Training set : $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$

m examples $\quad x \in \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \quad (\mathbb{R}^{n+1}) \quad x_0 = 1, \; y \in \{0, 1\}$

$h_\theta(x) = \dfrac{1}{1+e^{-\theta^T x}}$

How to choose $\theta$?

Cost function

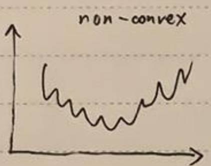
— Linear regression : $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \underbrace{\frac{1}{2}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2}_{cost(h_\theta(x^{(i)}),\, y^{(i)})}$

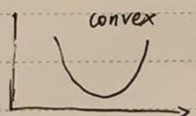
$cost(\underbrace{h_\theta(x^{(i)})}_{\text{prediction}}, \underbrace{y^{(i)}}_{\text{truth}})$

— Logistic regression : If we use the same definition ~~as~~ of cost function as linear regression. The cost function is NOT convex.

$cost(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2}(h_\theta(x^{(i)}) - y^{(i)})^2$ where

$h_\theta(x^{(i)}) = \dfrac{1}{1+e^{-\theta^T x^{(i)}}}$

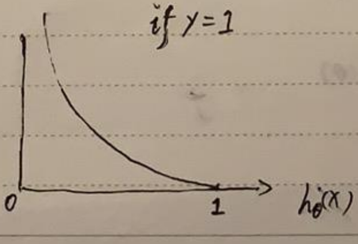



only 1 global max/min

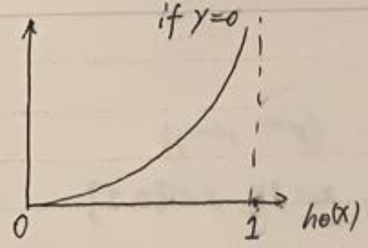
How to fix it?

$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$




Property : Cost $= 0$ if $y = 1$, $h_\theta(x) = 1$

But as $h_\theta(x) \to 0$, Cost $\to \infty$




Remember :

Logistic regression cost function

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}\left( h_\theta(x^{(i)}), y^{(i)} \right)$$

$$\text{cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y=1 \\ -\log(1-h_\theta(x)) & \text{if } y=0 \end{cases}$$

Note : $y=0$ or $1$ always

We can comprise them together

$$\Rightarrow \text{Cost}(h_\theta(x), y) = -y\log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$$

If $y=1$, the second term goes away.

So, $\text{Cost}(h_\theta(x), y) = -\log(h_\theta(x))$

If $y=0$, the first term goes away.

So, $\text{Cost}(h_\theta(x), y) = -\log(1-h_\theta(x))$

For all the examples :

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}\left( h_\theta(x^{(i)}), y^{(i)} \right)$$

$$= -\frac{1}{m} \left[ \sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)})) \right]$$

To fit parameters $\theta$:

$$\min_\theta J(\theta)$$

To make a prediction given new $x$:

Output $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \iff p(y=1 \mid x;\theta)$

Gradient Descent

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))\right]$$

Want $\min_\theta J(\theta)$

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

learning rate

$$\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})\, x_j^{(i)}$$

(Simultaneously update all $\theta_j$)

}.

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})\, x_j^{(i)}$$

Q: How to make it faster? A: Feature Scaling

Q: How to ganurantee the convergence? A: Backtracking

More optimization algorithms

1) Conjugate gradient
2) BFGS
3) L-BFGS

Ad: No need to manually pick $\alpha$; often faster

Dis: · more complex

理论推导: MLE

$P(y_i = 1 \mid x_i; \theta) = \theta^T x$

where $h_\theta(x) = g(\theta^T x) \Rightarrow \frac{1}{1+e^{-\theta^T x}}$

$P(y_i = 1 \mid x_i; \theta) = \frac{1}{1+e^{-\theta^T x}} = h_\theta(x)$

$P(y_i = 0 \mid x_i; \theta) = 1 - h_\theta(x)$

$P(y_i \mid x_i; \theta) = h_\theta(x_i)^{y_i}(1-h_\theta(x_i))^{(1-y_i)} \rightarrow$ Bernoulli

⇓ for all of the examples

$P(Y \mid X; \theta) = \prod_{i=1}^{m} h_\theta(x_i)^{y_i}(1-h_\theta(x_i))^{(1-y_i)} \Longrightarrow L(\theta)$ it allows us to find $\theta$

So, we want to $\max_\theta (l(\theta))$

$$L(\theta) = \prod_{i=1}^{m} h_\theta(x_i)^{y_i} (1 - h_\theta(x_i))^{(1-y_i)}$$

$$\ell(\theta) = \sum_{i=1}^{m} y_i \log(h_\theta(x_i)) + (1-y_i)\log(1 - h_\theta(x_i))$$

We want to maximize the $\ell(\theta)$ in order to make the model most ~~appa~~ plausible to represent the whole data.

Take one data as example

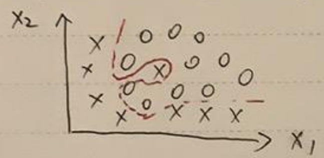
$$\frac{\partial \ell(\theta)}{\partial \theta} = \frac{y}{h_\theta(x_i)} \cdot \boxed{\frac{\partial h_\theta(x_i)}{\partial \theta}} + \frac{1-y_i}{1-h_\theta(x_i)} \cdot -\boxed{\frac{\partial h_\theta(x_i)}{\partial \theta}}$$

$$\left( \text{where } \frac{\partial h_\theta(x_i)}{\partial \theta} = x_i h_\theta(x_i)(1 - h_\theta(x_i)) \right)$$

$$= (y - h_\theta(\bar{x}))\,\bar{x}$$

$$\therefore \frac{\partial \ell(\theta)}{\partial \theta} = (y - h_\theta(\bar{x}))\,\bar{x}$$

Regularized logistic Regression




$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \cdots)$$

$\downarrow$

overfitting

$$J(\theta) = \left[ -\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{i}) + (1-y^{(i)}) \log(1 - h_\theta(x^{i})) \right] + \underbrace{\frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2}_{\text{regularization term}}$$

Even though the polynomial term is large, regularization makes $\theta$ small to avoid overfitting.

Gradient descent

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})\, x_0^{(i)} \rightarrow \theta_0 \text{单拿出来}$$

$$\theta_j := \theta_j - \alpha \underbrace{\left[ \frac{1}{m} \sum_{i=1}^{m} h_\theta(x^{(i)}) - y^{(i)})\, x_j^{(i)} - \frac{\lambda}{m} \theta_j \right]}_{\frac{\partial}{\partial \theta_j} J(\theta)} \quad j = 1, 2, \cdots n$$

# logistic regression example

Zijing Gao

### This data set contains the following features:

1. 'Daily Time Spent on Site': consumer time on site in minutes
2. 'Age': customer age in years 'Area Income': Avg. Income of geographical area of consumer
3. 'Daily Internet Usage': Avg. minutes a day consumer is on the internet
4. 'Ad Topic Line': Headline of the advertisement
5. 'City': City of consumer
6. 'Male': Whether or not consumer was male
7. 'Country': Country of consumer
8. 'Timestamp': Time at which consumer clicked on Ad or closed window
9. 'Clicked on Ad': 0 or 1 indicated clicking on Ad

```
#### Logistic Regression ####

# Load the data
path = "C:/Users/gzjgz/OneDrive/Documents/Advertising.csv"
```

```r
mydata = read.csv(path)
head(mydata)
```

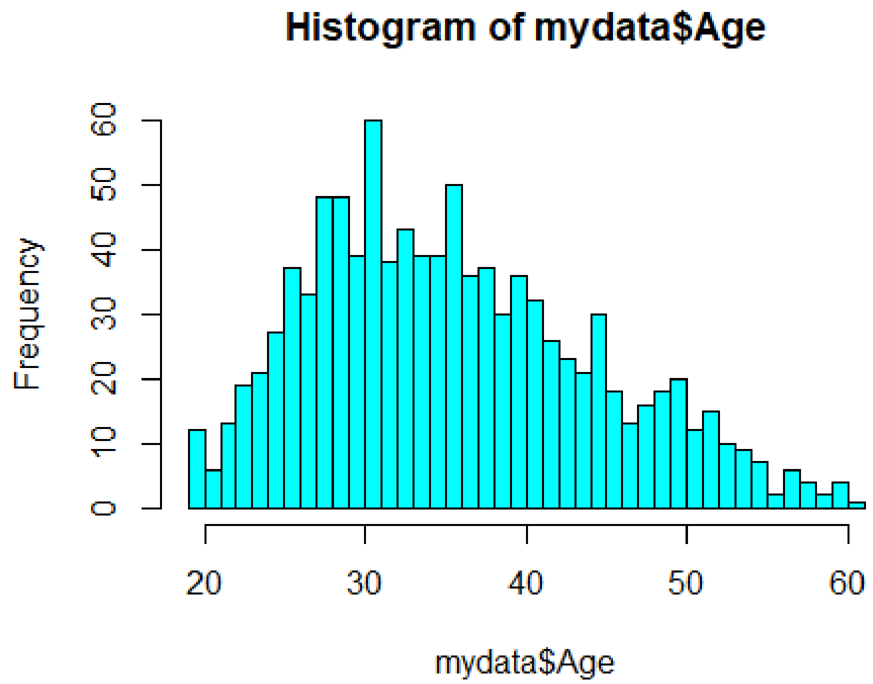
```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    68.95  35    61833.90               256.09
## 2                    80.23  31    68441.85               193.77
## 3                    69.47  26    59785.94               236.50
## 4                    74.15  29    54806.18               245.89
## 5                    68.37  35    73889.99               225.58
## 6                    59.99  23    59761.56               226.74
##                           Ad.Topic.Line           City Male    Country
## 1    Cloned 5thgeneration orchestration    Wrightburgh    0    Tunisia
## 2    Monitored national standardization      West Jodi    1      Nauru
## 3      Organic bottom-line service-desk       Davidton    0 San Marino
## 4 Triple-buffered reciprocal time-frame West Terrifurt    1      Italy
## 5         Robust logistical utilization   South Manuel    0    Iceland
## 6       Sharable client-driven software      Jamieberg    1     Norway
##             Timestamp Clicked.on.Ad
## 1 2016-03-27 00:53:11             0
## 2 2016-04-04 01:39:02             0
## 3 2016-03-13 20:35:42             0
## 4 2016-01-10 02:31:19             0
## 5 2016-06-03 03:36:18             0
## 6 2016-05-19 14:30:17             0
```

```r
# check null
sapply(mydata,function(x) sum(is.na(x)))
```

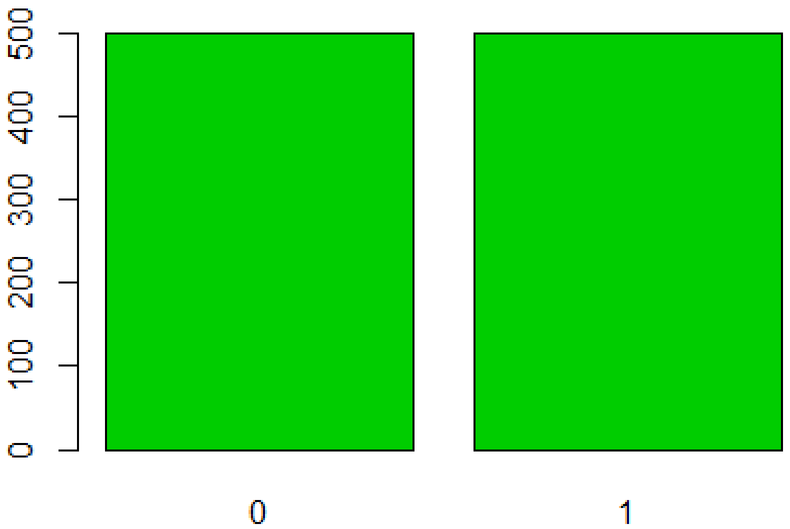
```
## Daily.Time.Spent.on.Site                      Age              Area.Income
##                        0                        0                        0
##     Daily.Internet.Usage            Ad.Topic.Line                     City
##                        0                        0                        0
##                     Male                  Country                Timestamp
##                        0                        0                        0
##            Clicked.on.Ad
##                        0
```

## EDA

```r
# hist plot of age
hist(mydata$Age,breaks = 30, col = 5)
```

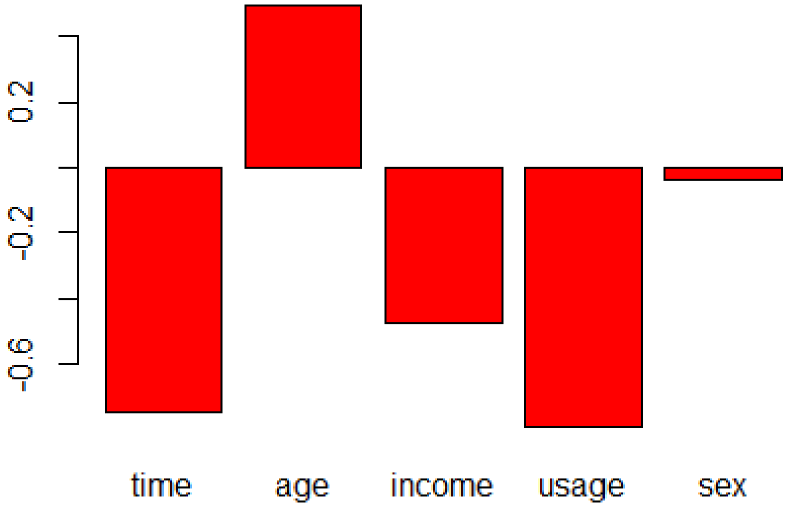



```r
# check if the target is balanced
barplot(table(mydata$Clicked.on.Ad), col = 3)
```




```r
# corelation with the target
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

numeric_data = select_if(mydata, is.numeric)
colnames(numeric_data) <- c("time", "age", "income", "usage", "sex", "click
")
corr = cor(numeric_data)[,6][1:5]
barplot(corr, col = 2)
```

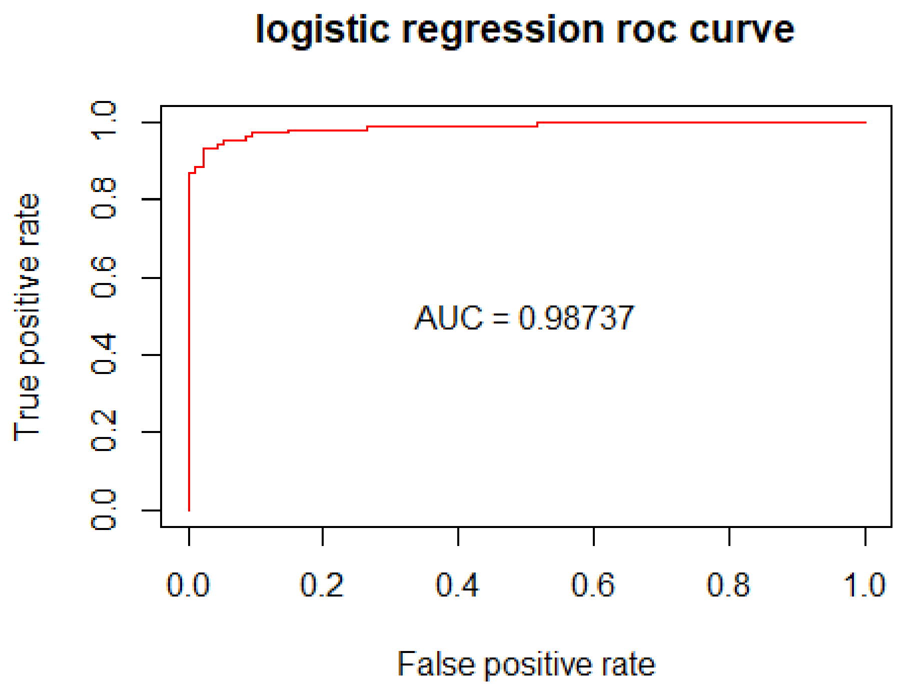



```
# train test split
# we drop the "sex" column since it is not corelated to the target.
X = subset(numeric_data, select = -c(sex))

X$click = factor(X$click)
train_idx = sample(nrow(X), 0.8*nrow(X))
train = X[train_idx,]
test = X[-train_idx,]
```

```r
# construct the model
model = glm(click~., family = binomial(link = "logit"), data = train)
# family = "binomial"

summary(model)

##
## Call:
## glm(formula = click ~ ., family = binomial(link = "logit"), data = trai
n)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3892  -0.1164  -0.0621   0.0119   3.2256
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.919e+01  3.362e+00   8.683  < 2e-16 ***
## time        -2.110e-01  2.612e-02  -8.077 6.62e-16 ***
## age          1.809e-01  3.119e-02   5.799 6.66e-09 ***
## income      -1.595e-04  2.427e-05  -6.571 4.99e-11 ***
## usage       -6.251e-02  7.889e-03  -7.923 2.32e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1108.91  on 799  degrees of freedom
## Residual deviance:  130.04  on 795  degrees of freedom
## AIC: 140.04
##
## Number of Fisher Scoring iterations: 8

# prediction with test data
fitted.results = predict(model,newdata=subset(test,select=-c(click)),type=
'response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)

# classification error
misClasificError <- mean(fitted.results != test$click)
print(paste('Accuracy',1-misClasificError))

## [1] "Accuracy 0.955"

# confusion matrix
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2

cm = confusionMatrix(factor(fitted.results), factor(test$click))
cm

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 93  7
##          1  2 98
##
##                Accuracy : 0.955
##                  95% CI : (0.9163, 0.9792)
##     No Information Rate : 0.525
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.91
##
##  Mcnemar's Test P-Value : 0.1824
##
##             Sensitivity : 0.9789
##             Specificity : 0.9333
##          Pos Pred Value : 0.9300
##          Neg Pred Value : 0.9800
##              Prevalence : 0.4750
##          Detection Rate : 0.4650
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.9561
##
##        'Positive' Class : 0
##

### ROC-AUC
library(ROCR)

## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```
p <- predict(model, newdata=subset(test,select=-c(click)), type="response
")
pr <- prediction(p, test$click)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf, col = 2)

auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]

text(0.5, 0.5, sprintf("AUC = %0.5f", auc))
title("logistic regression roc curve")
```




```
# BOUNS: svm
# construct the model
library(e1071)
model.svm = svm(click~., data = train, probability = TRUE)
# family = "binomial"

summary(model.svm)

##
## Call:
## svm(formula = click ~ ., data = train, probability = TRUE)
##
##
```

```
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  91
##
##  ( 45 46 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```
# prediction with test data
pred.svm = predict(model.svm, subset(test, select = -c(click)), probability
 = TRUE)
```

```
# classification error
misClasificError <- mean(pred.svm != test$click)
print(paste('Accuracy',1-misClasificError))
```

```
## [1] "Accuracy 0.95"
```

```
# confusion matrix
library(caret)
cm = confusionMatrix(factor(pred.svm), factor(test$click))
cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 91  6
##          1  4 99
##
##                Accuracy : 0.95
##                  95% CI : (0.91, 0.9758)
##     No Information Rate : 0.525
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8998
##
##  Mcnemar's Test P-Value : 0.7518
##
```

```
##          Sensitivity : 0.9579
##          Specificity : 0.9429
##       Pos Pred Value : 0.9381
##       Neg Pred Value : 0.9612
##           Prevalence : 0.4750
##       Detection Rate : 0.4550
## Detection Prevalence : 0.4850
##    Balanced Accuracy : 0.9504
##
##     'Positive' Class : 0
##
```

```
### ROC-AUC
pr <- prediction(attr(pred.svm,"probabilities")[,1], test$click)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf, col = 2)

auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]

text(0.5, 0.5, sprintf("AUC = %0.5f", auc))
title("SVM roc curve")
```